

**VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky**

**Experimentální pracoviště pro intrusivní hodnocení kvality  
hovoru**

**Experimental Testbed for Intrusive Assessment of Speech Quality**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

## Zadání bakalářské práce

Student: **Karolína Benešová**  
Studijní program: B2647 Informační a komunikační technologie  
Studijní obor: 2601R013 Telekomunikační technika  
Téma: Experimentální pracoviště pro intrusivní hodnocení kvality hovoru  
Experimental Testbed for Intrusive Assessment of Speech Quality

Zásady pro vypracování:

1. Metody hodnocení kvality řeči PESQ a POLQA
2. Asterisk a jeho využití jako automatu generujícího volání
3. Návrh a realizace pracoviště pro intrusivní měření kvality řeči
4. Vytvoření sady řečových vzorků s různým MOS

Seznam doporučené odborné literatury:

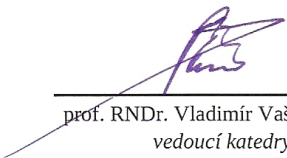
VOZŇÁK M. *Spojovací systémy*. VŠB-TU Ostrava, 1. vydání, 196 str., 2009, ISBN 978-80-248-1961-7.  
VOZŇÁK M. *Voice over IP*. VŠB-TU Ostrava, 1. vydání, 176 str., 2008, ISBN 978-80-248-1828-3.  
ROZHON J., BLAHA P., VOZNAK M., SKAPA J. *The Weather Impact on Speech Quality in GSM Networks*. SPRINGER: Communications in Computer and Information Science, Volume 291, 2012, pp. 360-369, DOI: 10.1007/978-3-642-31217-5\_38.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

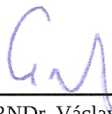
Vedoucí bakalářské práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

  
prof. RNDr. Vladimír Vašínek, CSc.  
vedoucí katedry



  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

Dne: 6.5.2013

*Bevešová*  
.....  
podpis studenta

## **Poděkování**

Ráda bych poděkovala Panu doc. Ing. Miroslavu Vozňákovi za odbornou pomoc a konzultaci při vytváření této diplomové práce. A dále také Ing. Janu Rozhonovi a Ing. Martinu Mikulcovi za rady ohledně konfigurace.

## **Abstrakt**

Tato práce se zabývá metodami hodnocení kvality hovoru a to převážně metodami intrusivními a jejich využití v experimentálním pracovišti. Pracoviště je založeno na IP telefonii pomocí software Asterisk a je realizováno na dvou stanicích. Tyto stanice mezi sebou komunikují, přičemž celá komunikace je nahrávána. Komunikace je simulována s variabilním zpožděním paketů a ztrátovostí přenosového kanálu. Vliv přenosového kanálu je simulován pomocí software Netem. U této komunikace je poté vyhodnocována výsledná kvalita pomocí volně dostupného PESQ algoritmu.

V teoretické části je pojednáváno o jednotlivých metodách hodnocení kvality hovoru a o software Asterisk. O jeho možnostech, vývoji a multifunkčnosti jako software pro vytváření pobočkových ústředen.

## **Klíčová slova**

Hodnocení kvality řeči, Asterisk, Netem, Bash, SIP, PESQ

## **Abstract**

This thesis deals with methods of assessing the quality of a speech mostly by intrusive methods and their application in an experimental workplace. The workplace is created on an IP telephony using Asterisk software and it is implemented at two PCs. The PCs communication between each other and entire communication is recorded. Communication is simulated with a variable delay and packet-loss caused by transmission channel. The effect of the transmission channel is simulated by using the software Netem. The final quality using freely available PESQ algorithm will then evaluated in this communication. A theoretical part of my work deals with a various methods of rating of speech quality and Asterisk software. Work also deals with its possibilities, development and multifunctionality as a software for building PBXs.

## **Key words**

Assessment of Speech Quality, Asterisk, Netem, Bash, SIP, PESQ

## Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
<b>ACD</b>	Automatic Call Distribution	Systém automatického volání
<b>ACR</b>	Absolute Category Rating	Absolutní hodnocení kvality
<b>CCR</b>	Comparision Category Rating	Porovnávací hodnocení kvality
<b>CLI</b>	Command Line Interface	Rozhraní příkazového řádku
<b>DCR</b>	Degradated Category Rating	Hodnocení pomocí degradace
<b>FIFO</b>	First-In-first-out	Typ fronty první přijde i odejde
<b>FFT</b>	Fast Fourier Transformation	Rychlá Fourierova transformace
<b>GPL</b>	General Public License	Veřejná licence
<b>IAX</b>	Inter Asterisk Exchange	Typ protokolu pro Asterisk
<b>IP</b>	Internet Protocol	Internetový protokol
<b>IVR</b>	Interactive Voice Response	Interaktivní hlasové menu
<b>MOS</b>	Mean Opinion Score	Stupnice pro hodnocení kvality
<b>MOS-CQ</b>	MOS-Conversation Quality	MOS konverzační kvalita
<b>MOS-LQ</b>	MOS-Listen Quality	MOS poslechová kvalita
<b>MOS-LQE</b>	MOS-Listen Quality Estimated	poslechová kvalita odhadovaná
<b>MOS-LQO</b>	MOS-Listen Quality Objective	poslechová kvalita objektivní
<b>MOS-LQS</b>	MOS-Listen Quality Subjective	poslechová kvalita subjektivní
<b>MOS<sub>LP</sub></b>	MOS listening-effort	Míra snahy pro porozumění
<b>MOS<sub>LE</sub></b>	MOS loudnesspreference	Hodnocení hlasitosti
<b>PBX</b>	Private Branch Exchange	Pobočková ústředna
<b>PC</b>	Personal Computer	Stolní počítač
<b>PESQ</b>	Perceptual Evaluation of Speech Quality	Procentuální vyjádření kvality hovoru
<b>POLQA</b>	Perceptual Objective Listening Quality Analysis	Procentuální, poslechová, objektivní kvalita hovoru

---

<b>RTP</b>	Real Time Protocol	Protokol pro přenos hlasu v IP telefonii
<b>SCCP</b>	Skinny Cisco Control Protocol	Cisco kontrolní protokol
<b>SIP</b>	Session Inicialization Protocol	Signalizační protokol
<b>TCP</b>	Transmission Control Protocol	Transportní protokol
<b>UDP</b>	User Datagram Protocol	Datagramový uživatelský protokol
<b>UTP</b>	Unshieled Twisted Pair	Nestíněná kroucená dvojlinka
<b>VoIP</b>	Voice over Internet Protocol	Internetová telefonie

---



# Obsah

1	Úvod.....	1
2	Hodnocení kvality hovoru.....	2
2.1	Subjektivní metoda.....	2
2.2	Objektivní metoda.....	3
2.2.1	Intrusivní metoda.....	3
2.3	PESQ.....	4
2.3.1	Algoritmus.....	4
2.4	POLQA.....	6
2.4.1	Rozdíly mezi PESQ a POLQA.....	6
2.5	MOS.....	8
3	Asterisk.....	10
3.1	Funkce Asterisku.....	10
3.2	Protokoly.....	12
3.3	Kodeky.....	13
3.4	Doplňkové funkce Asterisku.....	13
3.5	Dialplan.....	15
3.6	Režimy spouštění Asterisku.....	16
3.6.1	Příkazy v CLI.....	17
3.7	Automat generující volání.....	17
4	Experimentální pracoviště.....	18
4.1	Realizace pracoviště.....	18
4.1.1	Nastavení SIP peers.....	20
4.1.2	Nastavení Dialplanu.....	21
4.2	NETEM a PESQ.....	23
4.3	Koncepce pracoviště.....	25
5	Řečové vzorky.....	29

6	Závěr .....	30
	Použitá literatura .....	31
	Seznam příloh.....	xxxiii

---

# 1 Úvod

V dnešním technickém a uspěchaném světě lidé hojně využívají efektivního a rychlého způsobu komunikace. Jedná se o komunikaci pomocí rozvíjející se techniky. Patří zde emaily, sms a podobně, ale nikdy tyto technologie nepředběhnou komunikaci pomocí telefonních hovorů. Ty jsou oblíbeny díky možnosti rychlé domluvy. Tento způsob komunikace je sice efektivní, ale je omezován díky finanční náročnosti. Běžný uživatel tyto dopady pocítí méně než velká firma, která užívá telefon jako hlavní komunikační prostředek. Naštěstí je zde open source možnost nazývaná Asterisk, kterou začal v roce 1999 vyvíjet Mark Spencer. Jedná se o software, který je zcela zdarma a volně dostupný. Jediné náklady, jež jsou spojené s provozem. Asterisku je pořízení přístupu na internet nebo vybudovaná strukturovaná kabeláž. Pomocí Asterisku, lze využívat služeb pobočkové ústředny a to bez nutnosti placení poplatků, což uvítáme v případě dlouhých nebo mezinárodních hovorů. Asterisk využívá komunikace přes Internetový protokol. Volání může být realizováno pomocí počítače či pomocí IP telefonu. I přesto že je komunikace zdarma, požadují uživatelé určitou kvalitu hovoru. Kvalitu hovoru hodnotíme pomocí rozličných metod a může být odhadnuta před samotným hovorem, nebo vyhodnocena až po jeho ukončení.

Má bakalářská práce je založena na testování kvality hovoru po jeho ukončení pomocí intrusivní metody. V experimentu jsou simulovány nepříznivé vlivy přenosu signálu, mezi které patří například paketová ztrátovost a zpoždění. V průběhu měření byl pozorován vliv těchto parametrů na výslednou kvalitu hovoru. Simulace je realizována pomocí Asterisku, na kterém je probíhající komunikace nahrávána a poté je kvalita hovoru vyhodnocena pomocí volně dostupného PESQ algoritmu.

---

## 2 Hodnocení kvality hovoru

V dnešní době se v telekomunikacích rozvíjí přenos hovorových signálu pomocí technologií zdrojového kódování a následného paketového přenosu. Výhodou těchto technologií je snížení přenosové kapacity a také výrazně nižší provozní a pořizovací náklady, což se poté promítne i do konečné ceny hovoru. Snižování přenosové kapacity se provádí na základě kódování, které sníží celkovou kapacitu přenosu od třetiny až po čtvrtinu celkové přenosové kapacity. Tato technologie má také i své nevýhody, mezi které patří například zpoždění hovorového signálu a především zhoršení kvality samotného hovoru. Metody pro hodnocení kvality hovoru se primárně dělí na dvě skupiny: na subjektivní metody a objektivní metody. O těchto metodách je pojednáváno v doporučení ITU-T P.800 [4].

### 2.1 Subjektivní metoda

Subjektivní metody jsou založeny na názoru posluchače, nebo celé skupiny posluchačů. Musíme brát také ohled na to, že posuzování kvality je individuální. Stejný vzorek může být různými posluchači vyhodnocen s odlišnou kvalitou. A také je velmi pravděpodobné že při opakovaném poslechu téhož posluchače může být také vyhodnocená kvalita rozdílná.

U těchto metod musíme brát ohled také na podmínky hovoru a vlivy okolí:

- Vliv řečníka
  - kvalita hovoru – muž nebo žena, dialekt nebo spisovný jazyk, vady řeči
  - způsob mluvení – emoce, kadence mluvení, srozumitelnost
  - typ úlohy – čtený diktát, jednotlivá slova, dialog s počítačem
  - vliv prostředí
  - okolní šum
  - vzdálený hovor
  - ozvěna
- Vliv přenosového kanálu
  - odlišnost a zkreslení mikrofoni
  - vlastní šum kanálu
  - výpadky signálu
  - zkreslení kanálu

- Vliv působení ztrátových kodeků
  - mouth-to-ear zpoždění – zpoždění do 150 ms jsou většinou nepostřehnutelné, rozmezí 150-400 ms se objevují pomlky a nad 400 ms je zpoždění již zcela zřetelné
  - echo – jedná se o odrazení hovorového signálu zpět k hovořícímu účastníkovi, což se projevuje jako ozvěna.

Tyto metody můžeme ještě dále dělit na konverzační testy a poslechové testy. Testy konverzační, jak již z názvu vyplývá, jsou testy o vzájemné interakci obou posluchačů. Tato metoda je objektivnější než metoda poslechová, ale také časově náročná.

Testy poslechové probíhají jednostranně, kdy na jedné straně účastník čte daný řetězec slov a na druhé straně komunikačního řetězce je posluchač, který zapisuje to, co slyší ve sluchátku. Původní řetězec slov a řetězec napsaný na papíře se poté porovná a vyhodnotí se tím kvalita řeči.

Mezi nejznámější subjektivní metody patří:

- ACR (Absolute Category Rating), která je založena na přímém hodnocení, kdy každé rozmluvě je ihned přiřazeno hodnocení daným subjektem.
- DCR (Degradation Category Rating Metod) využívá hodnocení pomocí nedegradovaného a degradovaného vzorku.
- CCR (Comparsion Category Rating Metod) je obdobou metody CCR viz.[1]

U metod subjektivních se také užívá tzv. Logatom viz.[3]. Jedná se o nesmyslné spojení hlásek a tím vzniká slovo bez jakéhokoliv významu. Tyto slova jsou používána z důvodu vlivu lidského mozku, který je schopen si necelá slova domyslet nebo chybné hlásky přehlédnout. Proto když je slovo neznámé a nemá žádný význam je také zaručena větší soustředěnost posluchače.

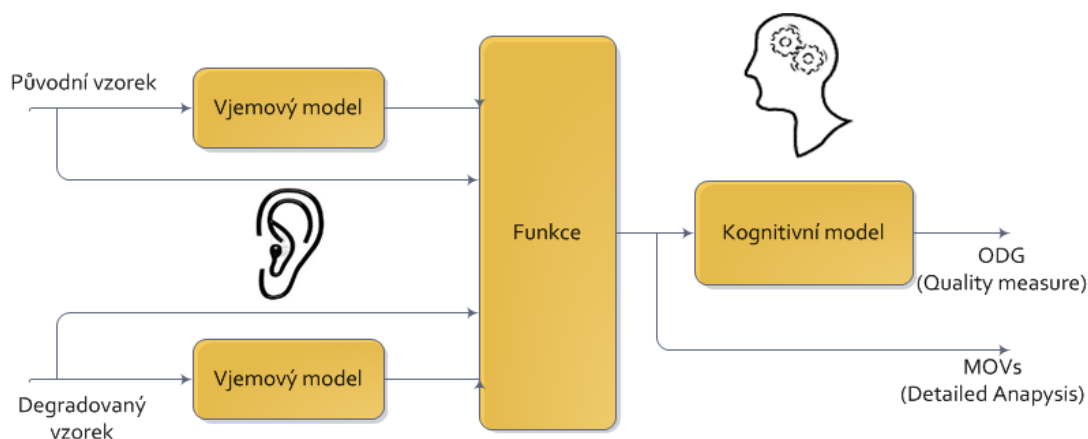
## 2.2 Objektivní metoda

Objektivní metody jsou založeny na nahrazení poslechového testu matematickým algoritmem. Tento algoritmus vyhodnocuje statisticky na základě matematických modelů, které více nebo méně modelují lidský sluchový aparát. Dále také vyhodnotí odhad předpokládaného subjektivního dojmu a tím i kvalitu hovoru. Tyto metody se dále dělí na neintrusivní metody a intrusivní metody.

### 2.2.1 Intrusivní metoda

Intrusivní metody jsou založeny na principu porovnání původního a degradovaného vzorku, který nám vznikne po průchodu přenosovým řetězcem. Tyto vzorky se porovnávají pomocí algoritmu,

který je odlišný podle metody, kterou si pro hodnocení vybereme. Obecné schéma metody je viditelné na Obrázku 2.1. Tato metoda je jednou z nejvhodnějších pro získání přesné kvality. Přesností se blíží k metodám subjektivním, ale její výhodou je, že není tak časově a finančně náročná. Má také svou nevýhodu. Protože potřebujeme původní i degradovaný vzorek, tak můžeme kvalitu vyhodnotit pouze zpětně, což je využitelné z hlediska výzkumu, ale v praxi se využijí spíše metody neintrusivní.



Obrázek 2.1: Schéma intrusivní metody

## 2.3 PESQ

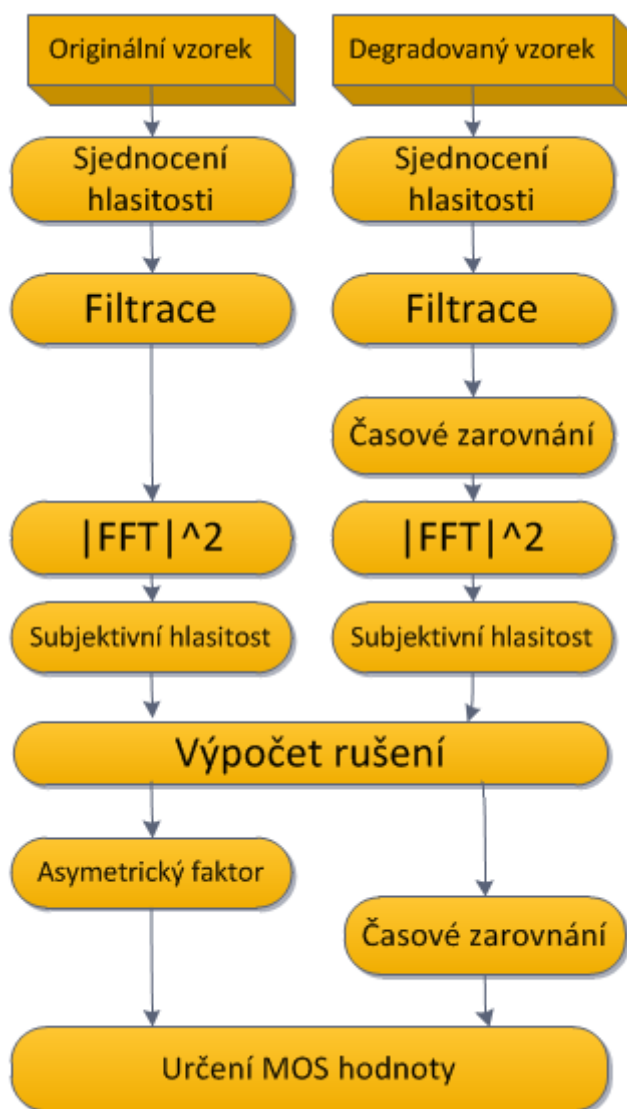
PESQ (Perceptual Evaluation of Speech Quality) je objektivní intrusivní metoda hodnocení kvality hovoru. Je definována v ITU-T P.862, kde také najdeme specifikaci a algoritmus této metody viz.[6]. PESQ je založena na principu porovnání původního vzorku  $X(t)$  a vzorku degradovaného  $Y(t)$ , který získáme po průchodu komunikačním řetězcem. Je nutné, aby se jednalo o reálný řečový vzorek, jinak může být vyhodnocená kvalita chybná. Je to proto, že vzorky umělé řeči nemusejí obsahovat časové a kmitočtové vlastnosti řeči reálné. Tato metoda je výsledkem několikaletého vývoje a používá se nejen pro zjištění zhoršení kvality způsobené kodeky hovoru, ale také pro měření end-to-end zpoždění. Algoritmus se snaží vystihnout ve zvukovém signálu důležité parametry, které ovlivňují vnímání kvality tohoto signálu. Tyto parametry jsou poté porovnávány u vzorku původního, tak u vzorku degradovaného. Na základě tohoto porovnání je poté odhadnut úbytek kvality a oba vzorky jsou klasifikovány pomocí stupnice MOS. Při zvýšené paketové ztrátovosti nejsou tyto odhady vždy přesné.

### 2.3.1 Algoritmus

Prvním krokem algoritmu je vyrovnání hlasitosti obou vzorků. Je důležité, aby oba vzorky měly stejnou úroveň, jež je rovna 79 dB. Dále následuje IRS filtrace, tato filtrace zohledňuje frekvenční vlastnosti telefonního zařízení. Aby se mohly vzorky porovnávat, musí se ještě provést časové zarovnání. Zarovnání je nutné, aby si odpovídaly jednotlivé úseky vzorku původního a

degradovaného. Nejdůležitější částí celého algoritmu je Fourierova transformace. Jedná se o časově-frekvenční transformaci, která slouží pro převod signálu z časové oblasti do oblasti frekvenční. Po transformaci je provedeno přizpůsobení vnímání lidského ucha a výpočet rušení, který je založen na rozdílech mezi hlasitostmi. Je-li rozdíl hlasitostí kladný, tak původnímu vzorku přidáme šum. Je-li rozdíl záporný, byl degradovaný vzorek zeslaben. Asymetrický faktor eliminuje změny frekvenčních složek, které vznikly po průchodu kodekem. Výsledná hodnota je vypočítána z průměrných hodnot. Přehledněji je celý algoritmus popsán na Obrázku 2.2.

Algoritmus PESQ je určen pro hodnocení kvality řeči o 3,1 kHz, což znamená, že jde o narrowband (úzkopásmové) a wideband (širokopásmové). Wideband pásmo se však objevuje až v pozdější verzi PESQ ITU-T P.862.2.



Obrázek 2.2: Vývojový diagram PESQ algoritmu

## 2.4 POLQA

POLQA (Perceptual Objective Listening Quality Analysis) je další generace testování kvality hovoru pro fixní, mobilní i IP telefony. Jeho vývoj probíhal od roku 2006 až do roku 2011 a stal se z něj další vývojový stupeň po metodě PESQ, viz. Obrázek 2.3 a [16]. Je využitelný pro všechna telefonní pásma, což je jeho nespornou výhodou oproti jeho předchůdci. Jde o intrusivní metodu, což znamená, že budeme porovnávat původní signál se signálem degradovaným. Algoritmus postupuje obdobně jako u metody PESQ, avšak jsou mezi nimi určité rozdíly, viz. Obrázek 2.4.

### 2.4.1 Rozdíly mezi PESQ a POLQA

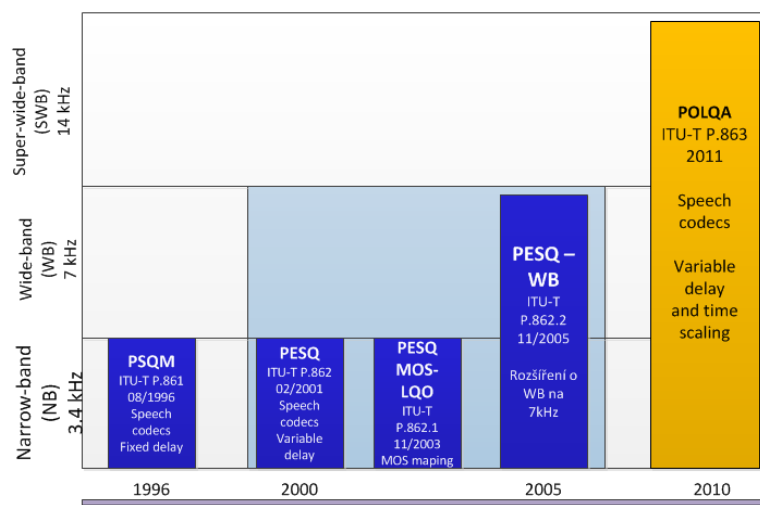
Hlavními rozdíly mezi těmito algoritmy jsou pásma, ve kterých můžeme tyto algoritmy užít. PESQ je určen pro narrowband a wideband pásma, kdežto POLQA je rozšířena o super-wideband, což umožňuje nahrávky nad 7 kHz, které mají vyšší kvalitu. Dalším rozdílem je, že pro ideální výsledek by PESQ vyžadoval MOS-LQO, kdežto u POLQA nejsou specifikovány přesnější požadavky. POLQA také řeší problém kdy výsledná MOS hodnota je nižší než bylo předpokládáno. Toto je způsobeno idealizací původního vzorku posluchačem, který má vzorek pouze ve své hlavě. POLQA před samotným porovnáním idealizuje vzorek originální, tedy odstraní ruchy a šumy a poté jej srovná s neidealizovaným vzorkem degradovaným. PESQ je také citlivé na rozdíly výkonových úrovní jednotlivých vzorků, kdežto u POLQA je přesněji definováno jaké úrovně by měly vzorky nabývat. U PESQ se také setkáváme s citlivostí na rozdíl vnitřních hodin u digitálně analogového převodníku oproti převodníku analogově-digitálnímu, tento problém je však v POLQA vyřešen. Největším rozdílem je výsledný rozsah MOS hodnot. Vyhodnocení samotných hodnot je založeno na typu algoritmu, proto jsou u obou metod mírně odlišné výsledné hodnoty viz. Tabulka 2.1.

Další rozdíly jsou obsaženy v [15].

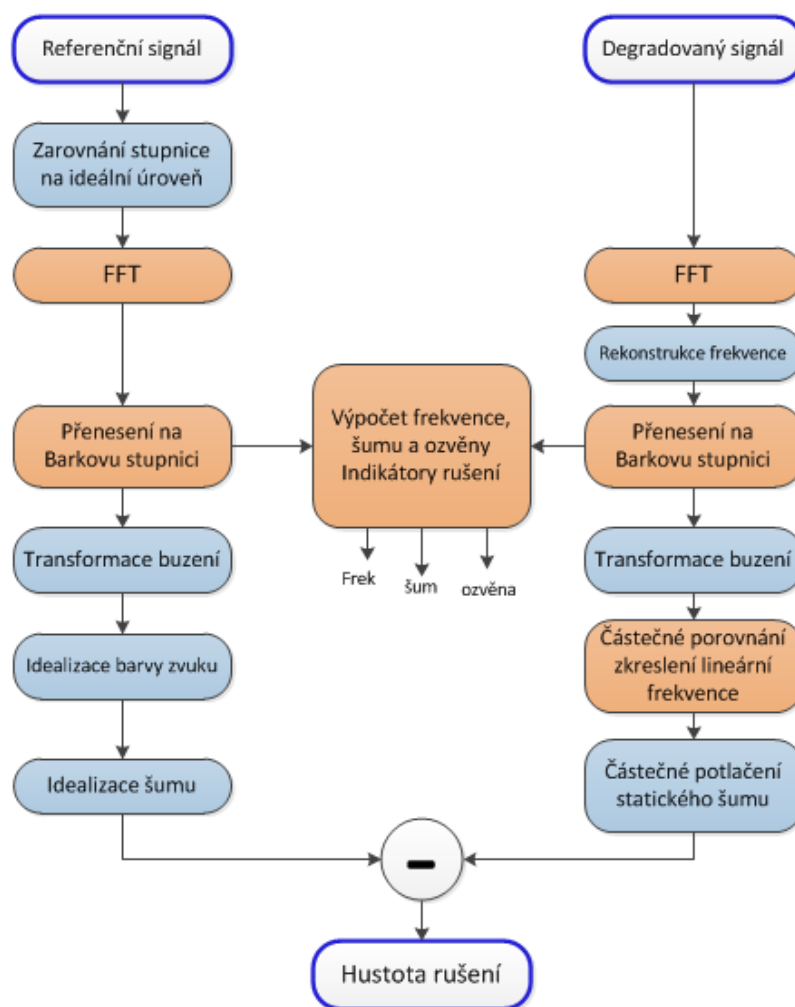
Pásmo (Mód)	PESQ MOS <sub>min</sub>	PESQ MOS <sub>max</sub>	POLQA MOS <sub>min</sub>	POLQA MOS <sub>max</sub>
Narrowband	1	4.5	1	4.5
Wideband	1	4.5	-	-
Super-wideband	-	-	1	4.75

Tabulka 2.1: Mezní hodnoty MOS





Obrázek 2.3: Vývoj intrusivních metod



Obrázek 2.4: Vývojový diagram algoritmu POLQA

## 2.5 MOS

MOS (Mean Opinion Score) je stupnice pro vyhodnocení kvality řeči. Tato stupnice je definována v ITU-T P.800 [5], kde nalezneme různé druhy stupnice, které jsou rozdílné podle potřeby testu. Všeobecně známou a nejvíce používanou je stupnice pětibodová. Na základě této stupnice, která obsahuje pět stupňů kvality, posluchač hodnotí kvalitu hovoru dle předem daných kritérií. Používá se číselného hodnocení z důvodu možnosti zprůměrování více posudků různých posluchačů. Takto můžeme získat přesnější a subjektivnější výsledky. Posluchači jsou předem proškoleni jak zní určitá kvalita vzorku podle MOS stupnice a poté mohou přistoupit k samotnému testu. Stupnice také urychluje testování. Nejenže zápis čísla kvality je rychlejší než slovní popis, ale jedná se také o standardizaci této metody, což umožňuje zabránit nejasnostem.

Stupnice postupuje od hodnoty pět, jakožto stupně nejlepší kvality, až po hodnotu jedna, která reprezentuje kvalitu nepřijatelnou. Stupnice je přehledněji zpracována v Tabulce 2.2.

Hodnocení	Kvalita	Quality	Zkreslení
5	Vynikající	Excellent	Rušení nerozpoznatelné
2	Dobrá	Good	Rušení rozpoznatelné, ale ne nepříjemné
3	Přijatelná	Fair	Rušení rozpoznatelné, mírně nepříjemné
2	Špatná	Poor	Rušení nepříjemné, ale ne protivné
1	Nepřijatelná	Bad	Rušení nepříjemné, protivné

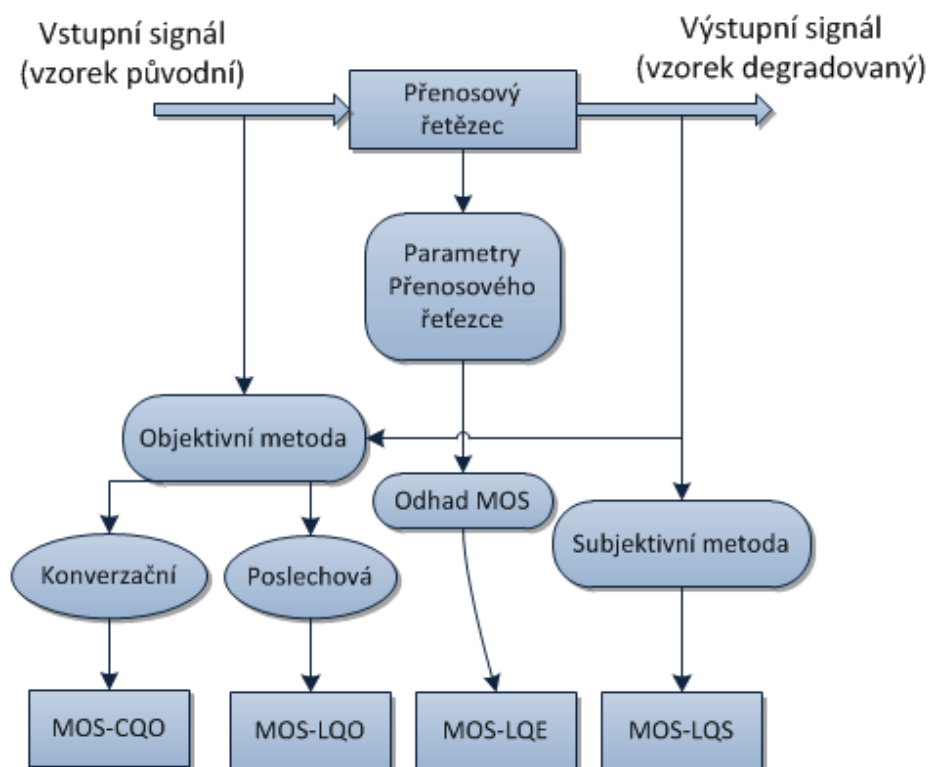
Tabulka.2.1: Stupnice MOS

Jako další stupnici MOS si můžeme představit stupnici MOS<sub>LE</sub> (MOS listening-effort). Ta představuje stupeň úsilí, které musí posluchač vynaložit, aby porozuměl hovoru. Další je stupnice MOS<sub>LP</sub> (MOS loudnesspreference), která specifikuje hodnocení spokojenosti s hlasitostí spojení a preferované hlasitosti.

Získaná hodnota ze stupnice MOS se liší na základě metody, jakou byl test prováděn. Jestliže byl test prováděn poslechovou metodou, jedná se o MOS-LQ (MOS-Listening Quality), v případě metody konverzační se jedná o MOS-CQ (Conversational Quality). Vztahy jednotlivých MOS hodnot jsou znázorněny na obrázku 2.5

MOS-LQ dále dělíme podle toho jakou metodou byl vzorek získán. MOS-LQS (MOS-Listening Quality Subjective) když byl vzorek získán subjektivní metodou. Dále MOS-LQO (MOS-Listening Quality Objective) když byl vzorek získán metodou objektivní. Vzorek také můžeme získat odhadem kvality. Jedná se o neintrusivní metodu, která je založena na monitorování probíhajících

spojení. Tato metoda se může provádět v reálném čase což je její nespornou výhodou. Avšak její nevýhodou je, že nemá k dispozici původní vzorek, proto je získaná kvalita MOS pouze odhadovanou a je pojmenována MOS-LQE (Listening Quality Estimated). Tuto odhadovanou kvalitu dostaneme například užitím E-modelu.



Obrázek 2.5: Vzájemný vztah mezi jednotlivými typy MOS

---

## 3 Asterisk

Asterisk začal vznikat již v roce 1999 a to zásluhou Marka Spencera, který projevil obrovské nadšení v tomto odvětví, jak pojednává [2]. Jeho snem bylo vytvořit privátní pobočkovou ústřednu PBX (Private Branch Exchange) pomocí open-source software, která by sloužila jako vhodná alternativa k již známým komerčním řešením PBX běžící pod licencí GPL (General Public License). Později se Mark spojil s Jimem Dixonem, který vyvíjel open-source hardware. Jejich spolupráce vytvořila vhodné prostředí pro open-source PBX jménem Asterisk. Později také založili firmu DIGIUM, jež se zabývá výrobou hardware pro Asterisk a z nabytých prostředků také financuje další vývoj Asterisku. Díky bezplatnosti a jednoduchosti je v dnešní době Asterisk jedním z nejrozšířenějších nástrojů pro PBX v IP telefonii co se týká open-source řešení. Jeho velkou výhodou je že není nutno pevně rezervovat linky pro každé koncové zařízení, lze je vytvářet dynamicky a jednotlivé koncové zařízení spolu mohou zdarma komunikovat a nejsou závislé na poskytovateli služeb. U firem můžeme využít výhody přenosu telefonie přes IP tak, že nemusíme vybudovávat ještě další síť telefonní vedle sítě počítačové.

Asterisk je napsán pomocí programovacího jazyka C a lze jej provozovat na operačních systémech Linux nebo Unix, které nabízejí rychlé a přehledné konfigurace v příkazovém řádku neboli Terminálu.

Dnes je vývoj Asterisku zajišťován díky open-source komunitě, což znamená že každý může přispět svým nápadem nebo inovací a tímto plně vyhovět požadavkům jednotlivých uživatelů. V roce 2012 vznikla zatím nejnovější verze Asterisku, verze 11. Podle oficiální webových stránek je na světě více než jeden milion PBX vytvořených pomocí Asterisku a to ve více než 170 zemích světa.

### 3.1 Funkce Asterisku

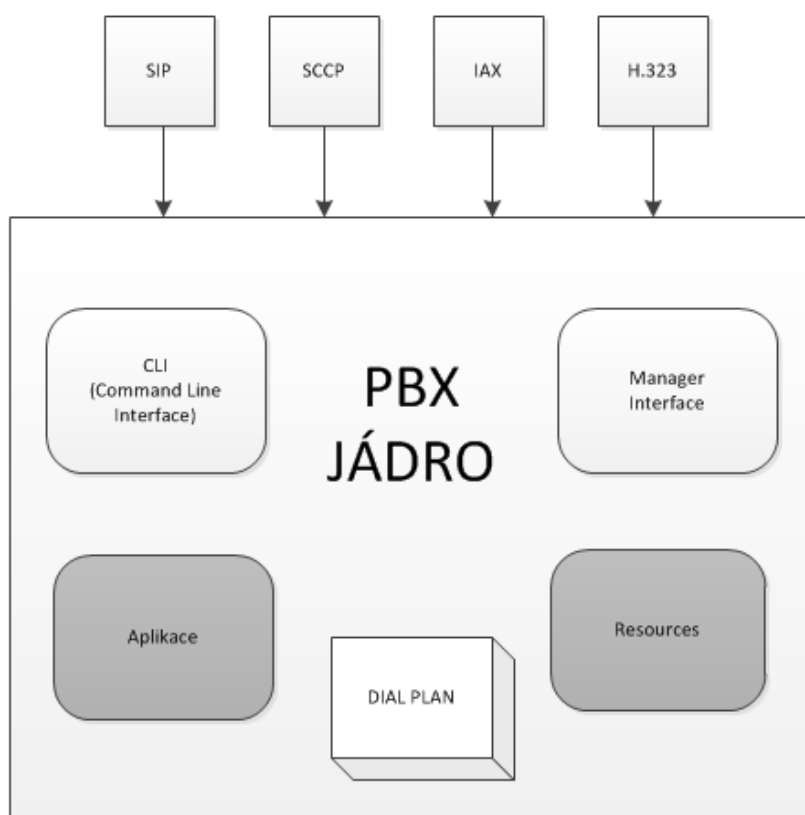
Asterisk je uzpůsoben pro řešení mnoha problémů ve VoIP telefonii, proto má několik různých účelů:

- Pobočková ústředna PBX
- Voicemail s jednotlivými službami i adresáři
- IVR server (Interaktivní hlasový průvodce) s možností distribuovaného automatického volání ACD (Automatic Call Distribution)
- Konferenční server
- VoIP gateway pro jednotlivé protokoly
- Softwarová ústředna

- Šifrování faxových i telefonních volání
- Možnost řadit hovory do front
- Překládání čísel
- Prediktivní volič
- využití jako vzdálené pracoviště pro již existující ústřednu

Asterisk už nyní dokáže poskytnout i služby, které byly dříve známy jen u pokročilých řešení pro firmy. Mezi tyto služby patří možnost zpříjemnit volajícímu čekací dobu předem zvolenou hudbou nebo melodií. Další funkcí určenou hlavně pro Call centra je možnost řadit hovory do fronty, kde k hovorům má přístup libovolný počet agentů a tyto hovory mohou být dále monitorovány. Možnost využití text-to-speech modulů, které dokážou daný text převést na mluvené slovo a také možnost rozpoznávání hlasu jednotlivých volajících. V neposlední řadě je zde možnost veškeré hovory probíhající na Asterisku převést do textové podoby a následně ukládat do databází.

Nejdůležitější částí samotného Asterisku je Dialplan, zde se definují všechny akce, které bude Asterisk provádět. Na obrázku 3.1 je znázorněna architektura software Asterisk.



Obrázek.3.1: Zjednodušená architektura Asterisku

## 3.2 Protokoly

Asterisk je také oblíben díky své rozmanitosti co se týče signalizačních protokolů. Tyto protokoly jsou nutné pro sestavení spojení a domluvu mezi jednotlivými koncovými zařízeními. Mezi nejznámější patří SIP, který je oblíben pro svou jednoduchost, což vyhovuje výrobcům hardware. Dále také IAX, který byl vytvořen speciálně pro Asterisk. Nesmíme zapomenout také na H.323 a především na Cisco protokol Skinny SCCP, který je rozšířen u větších korporací z důvodu užívání Cisco produktů. S těmito jednotlivými protokoly spolupracuje Asterisk pomocí jádra, které slouží jako rozhraní k jednotlivým kartám.

Protokol H.323 užíváme především pro konferenční hovory, protože je schopen zajistit přenos hlasový, obrazový i datový. Tento protokol však Asterisk nepodporuje v základní konfiguraci, ale díky rozšiřitelnosti Asterisku jej můžeme doinstalovat. Dnešním nástupcem se však stal protokol SIP.

Protokol SIP (Session Initialization Protocol) nebyl zprvu tak oblíben díky již existujícímu protokolu H.323. Svou nynější pozici si vybudoval hlavně díky otevřenosti svého kódu. Při užívání tohoto protokolu je jasné viditelné sestavení spojení. SIP je protokol aplikační vrstvy a pro jeho komunikaci je určen port 5060. SIP může být přenášén pomocí TCP i UDP protokolu transportní vrstvy. Pro přenos médií mezi koncovými stanicemi se používá RTP (Real Time Protocol) protokol. Mezi tyto média řadíme samotný telefonní hovor nebo videokonferenci.

Dalším protokolem vytvořeným speciálně pro Asterisk je protokol IAX. Jedná se o otevřený protokol, který však není standardem. Z důvodu standardizace se začal vyvíjet protokol IAX2, který však ještě není zcela dokončen. IAX je protokolem transportní vrstvy a pro svůj přenos a signalizaci využívá jeden UDP port 4569. Tento protokol je vhodný pro využití celé šířky pásma pokud posíláme několik hovorů najednou a to díky svému seskupování probíhajících relací do jednoho datového toku. Tento protokol je také nejmenší zátěží pro firewall, protože používá jenom jeden port UDP protokolu.

Cisco protokol SCCP (Skinny Cisco Control Protocol) je určen pro Cisco VoIP zařízení. Je především standardním protokolem pro koncová zařízení Cisco Manager PBX. Jednotlivé přiřazené porty a transportní protokoly pro jednotlivé protokoly jsou viditelné v Tabulce 3.1:

Protokol	Číslo portu	protokol transportní vrstvy
SIP	5060/5061	TCP/UDP
IAX2	4569	UDP
SCCP	2000	TCP
H323	1720	TCP
RTP	10,00-20,000	UDP

*Tabulka.3.1: Přidělení portů protokolům*

### 3.3 Kodeky

Kodeky užíváme neboť nám poskytují výhodné možnosti u digitálního přenosu hlasu. Jejich největší výhodou je komprese, která nám umožňuje vytvořit více hovorů najednou, a tím komunikační kanál maximálně využít. Dále nám kodek umožňuje detekovat hlasovou aktivitu a vyrovnat paketovou ztrátovost, což nám pomáhá získat lepší výslednou kvalitu hovoru. A v neposlední řadě umí kodeky generovat výplňový šum. Mezi jednotlivými kodeky je schopen Asterisk libovolně přecházet. Asterisk nám poskytuje využití těchto kodeků:

- G.711 alaw (Europe)
- G.711 ulaw (USA)
- G.722
- G.723.1
- G.726
- G.729
- iLBC
- LPC10
- Speex

### 3.4 Doplnkové funkce Asterisku

Aby se Asterisk vyrovnal profesionálním firemním PBX nabízí širokou škálu doplňkových funkcí, jak základních, tak i pokročilých. Zde jsou některé z nich:

- ACCOUNT CODE je funkce určená pro tarifování daného hovoru. Volající před samotnou volbou čísla zadá svůj kód. Pod tímto kódem je poté uložena délka hovoru, volané číslo, cena, atd.
- AUTOMATED ATTENDANT je využíváno pro přepojení bez účasti spojovatelky.
- AUTOMATIC HOLD tato funkce je velmi užitečná pro zaneprázdněné uživatele. Je-li potřeba provést další hovor a nechceme ukončovat hovor předešlý, Asterisk první hovor podrží a po vyřešení hovoru druhého se můžeme zpětně vrátit k hovoru prvnímu.
- BLACKLIST jde o seznam nevyžádaných čísel, které budou při příchozím hovoru zablokovány nebo odmítnuty.

- CALL DETAIL RECORD, záznam hovorů na námi vytvořené PBX. Tyto záznamy obsahují délku hovoru, číslo volajícího i volaného a datum
- CALL FORWARDING ON BUSY, je-li volané číslo obsazeno, je příchozí hovor automaticky přesměrován na daného uživatele.
- CALL FORWARDIN NO ANSWER, jako u předešlé funkce je číslo přesměrováno, ale na základě nevyzvednutí hovoru.
- CALL MONITORING, umožňuje zřídit evidenci příchozích, odchozích i zmeškaných hovorů. Do této evidence se uživatel dostane po autorizaci přes webové rozhraní.
- CALL QUEUING, je funkce vhodná pro Call centra. Umožňuje řadit hovory do fronty a ty pak mohou být vyzvednuty libovolným operátorem z dané skupiny.
- CALL RECORDING, je určeno pro nahrávání proběhlých hovorů. Tyto záznamy jsou ukládány v požadovaném formátu a poté jsou poskytnuty oprávněným uživatelům.
- CALL WAITING, upozorní na čekající hovor již během sestavení hovoru. Po přijetí hovoru je možné je libovolně střídat.
- DIAL BY NAME, místo telefonního čísla uživatele zadám jeho jméno
- DO NOT DISTURB, po aktivaci jsou hovory přesměrovány na spojovatelku, jiné pracoviště, atd.
- MUSIC HOLD ON, je určeno pro zpříjemnění čekání na přidružené lince pomocí hudby
- REMOTE CALL PICKUP, umožňuje vyzvednutí hovoru, který je určen pro jinou pobočku
- SMS MESAGING je rozšíření o SMS upozornění o zmeškaných hovorech, zanechaných vzkazech. Je nutno mít SMS bránu.
- TALK DETECTION, detekuje hovor ve spojení a dokonce rozezná osobu od záznamníku
- TIME AND DATE, funkce přečte volajícímu datum a čas
- VOICEMAIL, je určeno pro nahrání vzkazu volanému, které je poté zpřístupněn přes telefon nebo web. Je možno jej odeslat i jako email

Další doplňkové funkce jsou popisovány v [10]



### 3.5 Dialplan

Dialplan je srdcem celého Asterisku, v něm definujeme jaké příkazy má Asterisk vykonat při nastávajících situacích. Dialplan vytvořený v souboru `extensions.conf` je vždy statický, což znamená, že pro každé volání provede vždy stejné úkony. Syntaxe příkazu se skládá ze jména nebo čísla volaného, priority a funkce, která má být provedena.

**exten => číslo,priorita,příkaz(parametry) .**

Chceme-li příkaz zjednodušit, můžeme část příkazu nahradit pomocí příkazu `same` a poté následuje priorita a funkce. Můžeme využít také parametru `n` místo čísla priority. Parametr `n` určuje, že je hodnota priority vždy o jedna větší než v předešlém příkazu.

**same => priorita, příkaz(parametry) .**

Zde jsou uvedeny nejpoužívanější funkce dialplanu.

- **Goto()** nás přesune do požadované části dialplanu, které je pojmenováno shodně jako odkaz v závorce např. `Goto(Menu)` a přesuneme se do části dialplanu označené jako `[Menu]`.
- **Background()** stejně jako **Playback()** přehrává předem nahraný zvukový záznam. Je-li u `Playback()` během přehrávání zmáčknuta určitá klávesa, přehrávání skončí a nadále se provede úkon daný této klávese. `Background()` se užívá pro vytváření interaktivních hlasových menu.
- **WaitExten()** se vkládá do místa, kde čekáme na odezvu uživatele. Uživatel musí zvolit libovolnou DTMF volbu pro následné pokračování. `WaitExten()` s číslem v závorkách nám udává kolik sekund bude Asterisk čekat, než se na odezvu uživatele např. `WaitExten(5)` – systém počká 5 sekund a pak pokračuje dále v dialplanu.
- **Dial()** slouží k vytvoření telefonického spojení a je zde specifikováno se kterým uživatelem chceme spojení navázat.
- **Hangup()** Asterisk ukončí hovor pomocí zavěšení.
- **Echo()** funkce „otáčí“ hovor zpět na volající Asterisk, což znamená, že je volajícímu poslán do sluchátka jeho vlastní hlas.
- **Answer()** Asterisk vyzvedne hovor na daném čísle.
- **Monitor()** umožňuje nahrávání probíhající konverzace na Asterisku.

Po každé změně musí být dialplan aktualizován pomocí příkazu v CLI režimu

```
*CLI> dialplan reload
```

nebo pomocí příkazu v příkazovém řádku.

```
$ sudo /usr/sbin/asterisk -rx "dialplan reload"
```

Další návody pro konfiguraci jsou dostupné na [11],[12]a[13].

### 3.6 Režimy spouštění Asterisku

**\$ sudo asterisk** – Asterisk bez jakéhokoliv dalšího příkazu spustí Asterisk jako aplikaci na pozadí

**\$ sudo asterisk -h** – tento příkaz nám ukáže nápovědu a vlastnosti které můžeme užívat. Alternativou je také příkaz `man asterisk`

**\$ sudo asterisk -c** – spustí Asterisk na popředí a je vázaný na spojení. Když se uživatel odhlásí nebo ztratí spojení, Asterisk je vypnut. Tento režim je užíván pro testování, ale je nevhodný pro tvorbu. Pokud spustíme Asterisk v tomto režimu a nechceme již nadále v tomto režimu pracovat, opustíme jej pomocí `*CLI> core stop now`

**-v; -vv; -vvv;** atd

Tato volba může být použita i s jinými možnostmi (např.,-cvvv) s cílem zvýšit podrobnost konzoly výstupu. Funkce je stejná jako příkaz CLI základního nastavení podrobnosti `n`, kde `n` je libovolné celé číslo mezi 0 a 5 (celé číslo větší než 5 bude fungovat, ale nebude poskytovat žádné další podrobnosti). Někdy je užitečné nenastavovat podrobnosti vůbec. Například chceme-li zobrazit jenom chyby po spuštění, upozornění a varování.

**-d; -dd; -ddd;** atd se užívá obdobně jako `-v`, ale jeho výstup je v sestavovacím režimu (debug output). Je vhodné především pro vývojáře

**-r** je jeden s nepoužívanějších příkazů a užívá se pro připojení k `*CLI>`, když Asterisk běží jako daemon. Pro ukončení je určen příkaz `exit`

**-T** – tento režim přidá časovou známku k CLI výstupu

**-x** Umožňuje Asterisku předat řetězec jako kdyby byl napsán v `*CLI>`. Užívá se například pro rychlý výpis kanálu bez nutnosti spustit Asterisk v konzole pomocí příkazu `-rx 'core show channels'`

**-g** je určeno pro výpis core souboru, pokud dojde k chybě

### 3.6.1 Příkazy v CLI

**core show help** – zobrazí platné CLI příkazy

**core restart now** – ihned restartuje Asterisk. Ukončí CLI a vrátí se do linuxového řádku

**core stop now** – ihned zastaví Asterisk. Ukončí CLI a vrátí se do příkazového řádku

**sip reload** - obnovení a uložení nastavení u SIP peers

**sip show peers** – vypíše nakonfigurované SIP zařízení. Na výstupu zjistíme účet, jméno, zařízení a IP adresu

**dialplan reload** – důležitý příkaz při změně v souboru extensions.conf, aktualizuje Dialplan

**dialplan show** – zobrazí vše aktivní v Dialplanu. Zahrnuje nastavené akce v extensions.conf, ale není na ně omezena.

## 3.7 Automat generující volání

Asterisk jako automat generující volání je vhodný například pro experimenty s komunikačním řetězcem. Výhodou je automatický průběh experimentu, což znamená že není nutná přítomnost více osob, než osoby obsluhující. V případě mé bakalářské práce je automatu využíváno pro degradaci řečových vzorků pomocí funkcí Monitor() a Playback(). Dále lze také automat využívat pro automatické dotazování uživatelů na kvalitu. Kdy jsou otázky, které byly předem nahrány, přehrávány pomocí Playback() a odpovědi uživatelů nahrávány pomocí Monitor().

---

## 4 Experimentální pracoviště

Tato kapitola pojednává o pracovišti pro intrusivní metody hodnocení kvality hovoru. Pracoviště je realizováno pomocí software Asterisk a Netem, společně s využitím Bash skriptů fungujících pod operačním systémem Linux. Pro vyhodnocení kvality je užíváno PESQ algoritmu, který je volně dostupný v doporučení ITU-T P.862.

Původní myšlenka pracoviště bylo vytvořit virtuální operační systém Linux pomocí software VMware, který bude zastupovat jeden z reálných operačních systémů a bude komunikovat s operačním systémem umístěným v počítači. Pracoviště by poté bylo lehce přenositelné a mohlo by být realizováno na jakémkoliv PC po instalaci VMware playeru a nahrání vytvořeného obrazu operačního systému. Bohužel po sestavení celého pracoviště byla indikována chyba, která byla způsobena nedostatečnou výkonností zařízení, na kterém bylo pracoviště realizováno. Z tohoto důvodu bylo celé pracoviště realizováno mezi dvěma PC s operačními systémy Linux umístěnými v laboratoři.

### 4.1 Realizace pracoviště

Prvním krokem realizace pracoviště byla instalace PBX Asterisk, instalaci je nutno provést na obou PC, aby byla možná vzájemná komunikace. Instalace je prováděna přes terminál operačního systému Linux. Před samotnou instalací je nutno být superuser neboli root, toho dosáhneme pomocí příkazu:

```
sudo su
password:****
```

Jako heslo je zadáváno heslo nastavené pro operační systém Linux.

Poté je nutno vyhledat balíčky Asterisk a následně je instalovat. Vyhledání balíčků je provedeno pomocí příkazu

```
apt-get cache asterisk
```

a samotná instalace je realizována pomocí příkazu

```
apt-get install asterisk
```

Pokud při instalaci nastane problém, může to být způsobeno nepřítomností některého z potřebných linuxových balíčků, proto pomocí příkazu

```
apt-get update
```

provedeme aktualizaci, po provedení aktualizace je instalaci nutno opakovat.

Po úspěšné instalaci je software Asterisk spuštěn pomocí příkazů

```
asterisk
```

```
asterisk -r
```

pomocí kterých se dostaneme do prostředí příkazového řádku Asterisku CLI

Po dokončení instalací na obou PC je možno začít realizovat pracoviště. Na pracovních PC je nutno propojit síťové karty pomocí kříženého UTP kabelu a poté nastavit zvolené IP adresy na aktivní rozhraní. Pro PC1 byla určena IP adresa 158.196.81.105 a pro PC2 158.196.81.104. Nastavení adres je provedeno pomocí příkazů

```
ifconfig eth3 158.196.81.105/24 na PC1
```

```
ifconfig eth3 158.196.81.104/24 na PC3
```

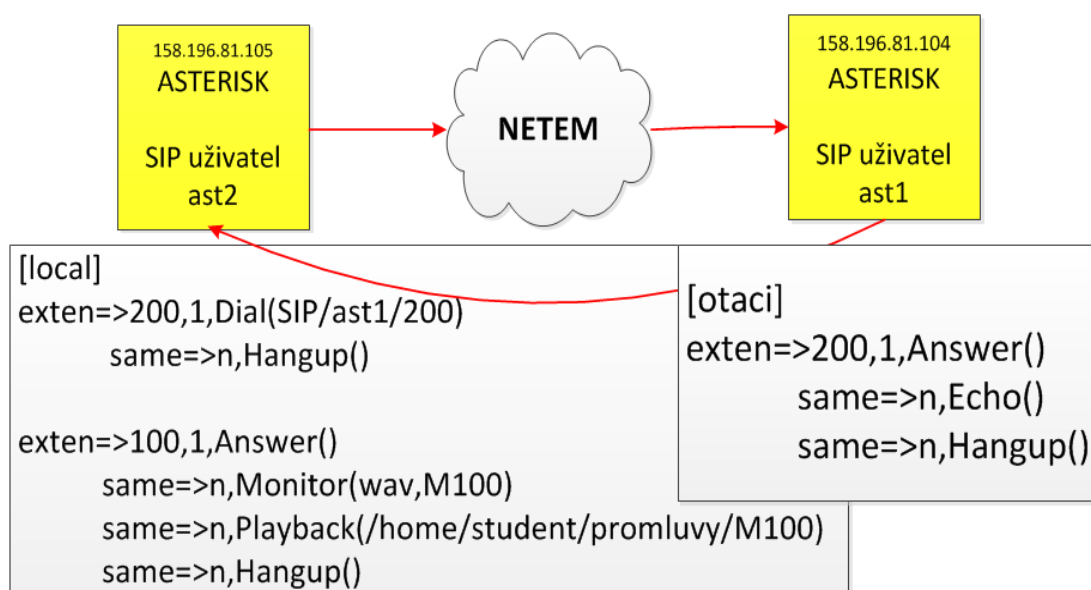
eth3 je označení aktivního rozhraní, kterým jsou PC propojeny, rozhraní je zjišťováno pomocí příkazu `ifconfig`, který vypisuje všechna aktivní rozhraní na daném PC.

Schéma pracoviště je znázorněno na obrázku 4.1.

Po úspěšném nastavení adres je nutno ověřit viditelnost PC mezi sebou. To ověříme pomocí příkazů

```
ping 158.196.81.104 z PC1
```

```
ping 158.196.81.105 z PC2
```



Obrázek.4.1: Schéma sestavovaného pracoviště

#### 4.1.1 Nastavení SIP peers

Pro komunikaci mezi Asterisky je nutno nastavit SIP uživatele. Nastavení probíhá v konfiguračním souboru `sip.conf`, který je umístěn v adresáři `/etc/asterisk`. Do konfiguračního souboru je třeba specifikovat IP adresu PC, dále na jakém portu bude komunikace probíhat, který kodek je preferován a také je nutno povolit transport UDP protokolu z důvodu komunikace pomocí RTP protokolu. Tyto údaje jsou nastavovány v části konfiguračního souboru `[general]`. V druhé části konfiguračního souboru je nastavován uživatel druhého PC. Je zde potřeba uvést IP adresu PC2, specifikovat typ uživatele jako `friend`, což znamená, že uživatel je schopen hovory nejen přijímat, ale i uskutečňovat. Dále je možno také uvést heslo. Pro autorizaci je nutno nastavit defaultní uživatelské jméno. Na závěr konfiguračního souboru je povoleno `qualify`, což zajišťuje průběžnou kontrolu dostupnosti klientů. Asterisk bude periodicky posílat SIP OPTIONS zprávu a kontrolovat zda je uživatel online. Zde je ukázka jednotlivých konfiguračních souborů `sip.conf`

`sip.conf` pro PC1:

```
[general]
port=5060
transport=udp
udpbindaddr=158.196.81.105
disallow=all
allow=alaw
```

```
[ast1]
host=158.196.81.104
type=friend
secret=1234
defaultuser=ast2
qualify=yes
```

`sip.conf` pro PC2:

```
[general]
port=5060
transport=udp
udpbindaddr=158.196.81.104
disallow=all
allow=alaw
```

```
[ast2]
host=158.196.81.105
type=friend
secret=1234
defaultuser=ast1
qualify=yes
```

---

Vkládání informací do souboru je možno provést příkazem

```
cd /etc/asterisk | cat > sip.conf,
```

který je určen pro přesun do požadované složky a následné zapisování do souboru sip.conf. Dále je také možno přistupovat do souboru pomocí příkazu `nano sip.conf`, nebo pomocí jiných textových editorů jako je například `vim` nebo `pico`

Po nastavení uživatelů spustíme Asterisk pomocí příkazu `asterisk -r` a zadáme příkaz `sip reload` a následně `sip show peers` pro výpis nastavených uživatelů.

#### 4.1.2 Nastavení Dialplanu

Dialplan je nejdůležitější částí software Asterisk. Zde je definováno jaké operace budou probíhat po vytočení daného čísla. Dialplan je umístěn v konfiguračním souboru `extensions.conf` defaultně umístěném v adresáři `/etc/asterisk`. Akce Asterisku jsou koncipovány v tomto tvaru

```
exten => číslo,priorita,příkaz(parametry),
```

tedy že po vytočení daného čísla je proveden tento nadefinovaný příkaz. Priorita určuje pořadí vykonávaných příkazů. Nebo jej lze nahradit příkazem `same => n,příkaz(parametry)`, kde slovo `same` nahrazuje část vytočení čísla a parametr `n` určuje prioritu vždy o jedno vyšší než u příkazu předchozího. V případě tohoto pracoviště budeme nastavovat na PC1 kontext `local`, který bude obsahovat vytočení čísla 200, které je určeno pro navázání spojení s Asteriskem v PC2 a po skončení celé akce spojení také ukončí. Tyto akce jsou prováděny pomocí příkazů `Dial()` a `Hangup()`. Další částí kontextu `local` je vytáčení čísel od 100 a výše. Počet volaných čísel je určen počtem vzorků, které jsou testovány. V našem případě kdy je testováno 25 vzorků je Dialplan nastaven na volání do čísla 124. Po vytočení čísla 100 je hovor vyzvednut pomocí příkazu `Answer()`. Po zvednutí následuje nahrávání komunikace probíhající při spojení. Příkazem `Monitor()` zajistíme uložení odchozího hovoru i hovoru příchozího. V parametrech nahrávání je určeno v jakém formátu a pod jakým názvem je nahrávka ukládána. Názvy nahrávek jsou voleny s ohledem na volané číslo pro následné využití v algoritmu testování. Nahrané záznamy se defaultně ukládají do adresáře `/var/spool/asterisk/monitor` a jsou ukládány ve tvaru `out-nazev.wav` pro odchozí hovor a `in-nazev.wav` pro hovor příchozí. Po zahájení nahrávání je nutno přehrát testovaný vzorek. Tohoto dosáhneme pomocí příkazu `Playback()` a zadání cesty k přehrávanému wav souboru. Funkce `Playback` akceptuje wav soubory pouze s vzorkovací frekvencí 8000Hz. Po skončení nahrávání příchozího hovoru je spojení ukončeno. Tento postup je aplikován i u ostatních vytáčených čísel s rozdílným názvem, pod kterým je nahrávka nahrávána a také rozdílným přehrávaným vzorkem.

Zde je ukázána část popisovaného konfiguračního souboru extensions.conf na PC1:

```
[local]
exten=>200,1,Dial(SIP/ast1/200)
    same=>n, Hangup()

exten=>100,1,Answer()
    same=>n, Monitor(wav,M100)
    same=>n, Playback(/home/student/promluvy/M100)
    same=>n, Hangup()

exten=>101,1,Answer()
    same=>n, Monitor(wav,M101)
    same=>n, Playback(/home/student/promluvy/M101)
    same=>n, Hangup()
```

Na PC2 je v Dialplanu context „otaci“, ve kterém je definováno vyzvednutí příchozího hovoru z čísla 200 pomocí Answer(). Poté následuje příkaz Echo(), který otáčí hovor zpět k volajícímu. Přesnějším vysvětlením příkazu Echo() je, že vše co je slyšeno v imaginárním sluchátku na PC2, je posláno i do pomyslného sluchátka PC1. Po skončení nahrávání na straně PC1 je spojení ukončeno. Zde je ukázka extensions.conf na PC2:

```
[otaci]
exten=>200,1,Answer()
    same=>n, Echo()
    same=>n, Hangup()
```

Po nastavení obou Dialplanu je nutno Dialplany aktualizovat v CLI příkazovém řádku pomocí příkazu dialplan reload. Můžeme si Dialplany také nechat zobrazit v terminálu pomocí příkazu dialplan show.

Po nastavení SIP uživatelů a Dialplanu ověříme správnost nastavení pomocí zkušebního volání. To provedeme v CLI pomocí příkazu:

```
originate local/100@local extenesion 200@local
```



Po zadání příkazu se nám v samotném CLI řádku zobrazují procedury, které jsou prováděny, popřípadě vzniklé chyby, jak je viditelné na obrázku 4.2.

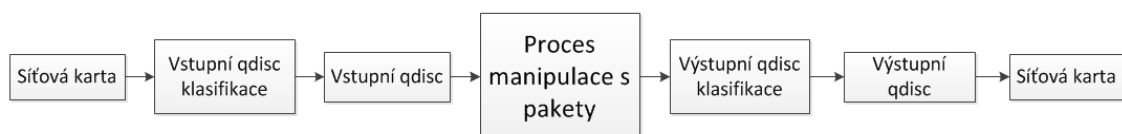
```
-- Executing [122@local:1] Answer("Local/122@local-c39a;2", "") in new stack
-- Remote UNIX connection disconnected
-- Executing [200@local:1] Dial("Local/122@local-c39a;1", "SIP/ast1/200") in
new stack
== Using SIP RTP CoS mark 5
-- Called SIP/ast1/200
-- Executing [122@local:2] Monitor("Local/122@local-c39a;2", "wav,M122") in
ew stack
-- Executing [122@local:3] Playback("Local/122@local-c39a;2", "/home/student
promluvy/M122") in new stack
-- <Local/122@local-c39a;2> Playing '/home/student/promluvy/M122.slin' (lang
age 'en')
-- SIP/ast1-000003f5 answered Local/122@local-c39a;1
-- Executing [122@local:4] Hangup("Local/122@local-c39a;2", "") in new stack
== Spawn extension (local, 122, 4) exited non-zero on 'Local/122@local-c39a;2'
== Spawn extension (local, 200, 1) exited non-zero on 'Local/122@local-c39a;1'
-- Remote UNIX connection
```

Obrázek. 4.2: Probíhající komunikace v CLI

## 4.2 NETEM a PESQ

Netem je síťový emulátor dostupný od verze Linux kernel 2.6.7 a ve verzích vyšších. Netem poskytuje možnost zpoždění, ztrátovosti, zahození, duplikaci a deformaci paketů. Netem je rozšíření tc neboli Linux traffic control, které je součástí iproute2 balíčku.

Traffic control je založeno na principu řízení cesty paketů na určitém rozhraní. Paket je vložen na NIC (network interface card), poté je vložen do fronty a klasifikován. Toto je prováděno před samotnou nastavenou akcí jako je zpoždění, ztrátovost a mnoho dalších, které provádí Linux. Po manipulaci je paket klasifikován a vložen do fronty pro další zpracování pomocí odchozího rozhraní síťové karty. Třídění paketů lze provádět na základě analýzy hlavičky paketu, která obsahuje cílovou a zdrojovou IP adresu, číslo portu a podobně, nebo pomocí obsahu. Třídění může být nastaveno pomocí kontroly provozu tc. Pakety jsou vkládány do vstupních nebo výstupních front. Celý proces je realizován pomocí postupu na Obrázku 4.3 Při standardní konfiguraci jsou tyto fronty realizovány zvlášť na rozhraní a jsou typu FIFO (first-in-first-out). Kombinace fronty a využitého algoritmu dále určuje paket, který bude klasifikován jako qdisc (krátký pro vložení do fronty).



Obrázek. 4.3: Schéma procesu traffic control

### Terminologie traffic control

- Queueing discipline – paket je vložen do fronty a algoritmus určuje kdy je odeslán
- Classless qdisc – qdisc bez nastavitelné vnitřní sekce
- Classful qdisc – qdisc, který může obsahovat třídy a podporuje klasifikaci paketů
- Root qdisc - je připojen ke každému rozhraní
- egress qdisc – pracuje na výstupním rozhraní
- ingress qdisc – pracuje na vstupním rozhraní
- Class – je připojena k qdisc a může obsahovat další třídy
- Filter – klasifikace může být provedena pomocí filtrace

### Základní operace NETEM

- delay – zpoždění pro každý paket
- loss – zahodí určité procento paketů
- duplication – duplikuje procento paketů
- corruption – poškození určitého procenta paketů

Ve standardním módu může být zpoždění definováno konstantou, odchylkou nebo vzájemným vztahem. Ostatní tři funkce lze specifikovat procenty nebo vzájemným vztahem.

V první části příkazu je třeba zvolit zda-li má být qdisc na rozhraní vytvořen, změněn, přesunut nebo odkazován. Dále je nutno určit na které rozhraní má být funkce aplikována a poté vybrat typ a hodnotu samotné funkce:

```
tc qdisc [ add | change | replace | link ] dev rozhraní [ parent  
název | root ] netem [loss|delay|duplication|corruption] [  
nastavovaná hodnota v ms nebo % ]
```

Jako příkladem je uvedena aplikace ztrátovosti 3% na rozhraní eth3

```
tc qdisc change dev eth3 root netem loss 3%
```

V případě pracoviště byl volen rozsah zpoždění 0 až 400ms s krokem 10ms a paketová ztrátovost s rozsahem 0 až 10% a krokem 0,5%.

Další příklady nastavení jsou uvedeny viz.[9]

PESQ algoritmus je volně dostupný online v doporučení ITU-T P.862 [9] ve formě zip souboru. Po získání souboru jej dekomprimujeme a adresář source zkopírujeme do adresáře

/var/spool/asterisk/. Soubor obsahuje zdrojové a hlavičkové soubory. V terminálu Linux se nacházíme v dané složce source a pomocí příkazu

```
gcc -o pesq *.c -lm
```

vytvoříme spustitelný binární soubor, který spouštíme z místa archívu obsahující porovnávané vzorky. Příkaz obsahuje cestu k binárnímu pesq souboru a vybranou vzorkovací frekvenci. Možnosti vzorkovací frekvence jsou 8000Hz nebo 16000Hz. Poslední částí příkazu je název referenčního a degradovaného vzorku. V tomto případě bude příkaz ve formě

```
cd /var/spool/asterisk/monitor
```

```
cd /var/spool/asterisk/source/pesq +8000 M100-out.wav M100-in.wav
```

Po zadání příkazu začne být v okně terminálu viditelný průběh algoritmu. Poté se zobrazí vyhodnocené MOS hodnoty viz Obrázek 4.4 a výsledky jsou průběžně ukládány do textového souboru pesq\_results.txt, který se automaticky vytváří v adresáři se vzorky, v našem případě /var/spool/asterisk/monitor.

```
*****
* OPTICOM GmbH                                * Psytechnics Limited                        *
* Naegelsbachstr. 38,                        * Fraser House, 23 Museum Street, *
* D- 91052 Erlangen, Germany                 * Ipswich IP1 1HN, England          *
* Phone: +49 (0) 9131 53020 0                 * Phone: +44 (0) 1473 261 800        *
* Fax:   +49 (0) 9131 53020 20                * Fax:   +44 (0) 1473 261 880        *
* E-mail: info@opticom.de,                   * E-mail: info@psytechnics.com,      *
* www.opticom.de                             * www.psytechnics.com               *
*****

Reading reference file M124-out.wav...done.
Reading degraded file M124-in.wav...done.
Level normalization...
IRS filtering...
Variable delay compensation...
Acoustic model processing...

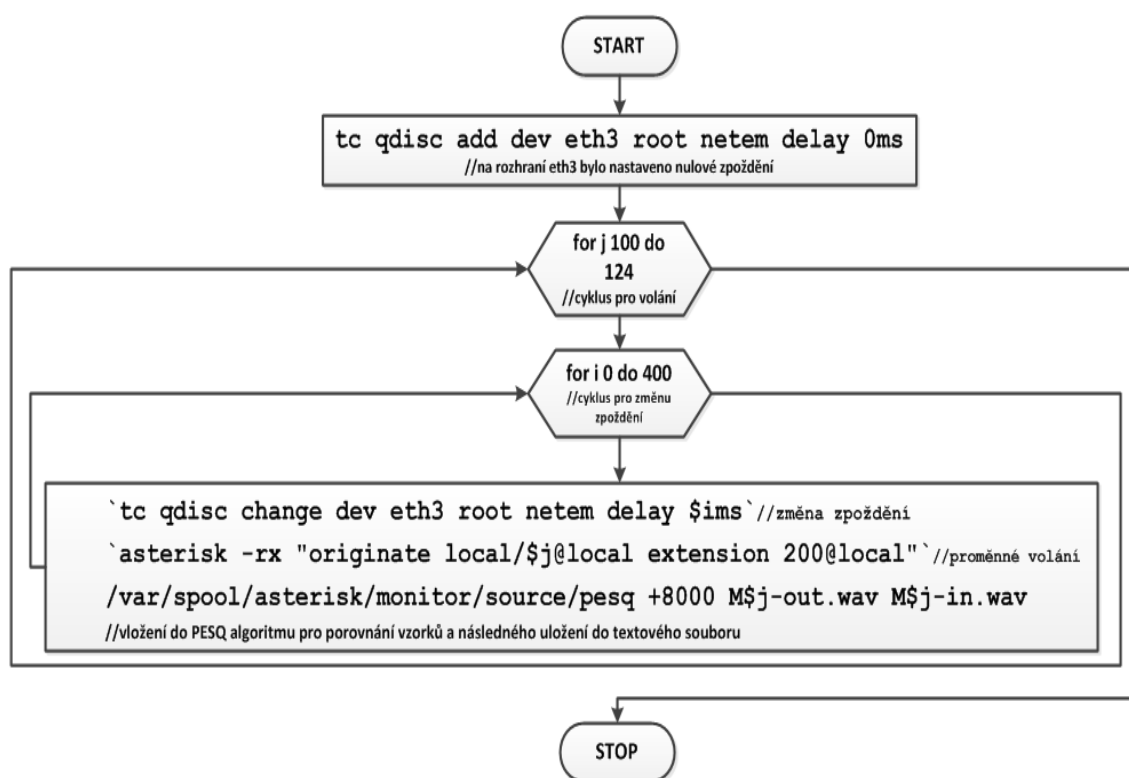
P.862 Prediction (Raw MOS, MOS-LQO):  = 3.449   3.484
```

Obrázek. 4.4: Ukázka výstupu z algoritmu

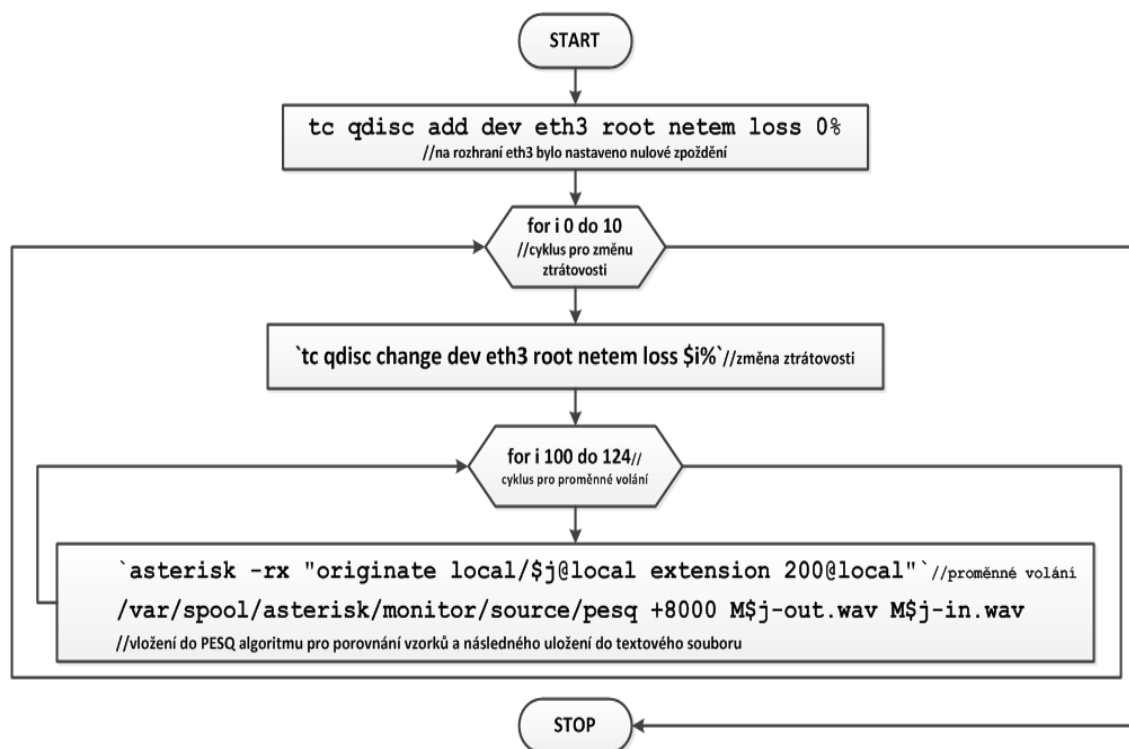
### 4.3 Koncepce pracoviště

Pracoviště je realizováno na bázi Linux bash skriptů, které fungují pomocí jednoduchých cyklů. Pro testování jednotlivých vzorků bylo nutno vytvořit cyklus pro degradaci všech vzorků, což znamená, že se musela postupně zavolat všechna čísla. To je zjištěno pomocí cyklu `for`, který pomocí volání proměnné zajistí průchod všech vzorků. Abychom zajistili proměnnou paketovou ztrátovost nebo zpoždění, musí být vytvořen cyklus druhý. Toto také zajišťuje cyklus `for`, který průběžně mění hodnoty. Po provedení degradace jednoho vzorku je vzorek ihned vložen do PESQ algoritmu a

porovnán se vzorkem originálním. Výsledek je zobrazen v okně terminálu a také uložen do textového souboru. Všechny skripty jsou umístěny v adresáři /var/spool/asterisk/monitor a to z důvodu vkládání porovnávaných vzorků. Pracoviště může fungovat dvěma způsoby. Při první variantě budeme nejdříve provádět všechny kroky ztrátovosti nebo zpoždění u jednoho vzorku a teprve poté se přesuneme na vzorek druhý. Ve variantě druhé nejdříve se stejnou hodnotou ztrátovosti či zpoždění zavoláme všechna čísla a teprve poté je hodnota navýšena. Vývojové diagramy těchto variant jsou viditelné na Obrázku 4.5 a Obrázku 4.6. Je třeba zvážit pro co budou výsledné hodnoty užívány. První způsob je vhodný pro následné zpracování jednotlivých vzorků a pozorování vlivu ztrátovosti nebo zpoždění. Způsob druhý je vhodný pro vzájemné porovnávání vzorků.



Obrázek. 4.5: Vývojový diagram varianty první



Obrázek. 4.6: Vývojový diagram druhé varianty

Do skriptů mohou být také umístěny výstupy z okna terminálu pro kontrolu probíhajícího skriptu. V případě skriptů pro zpoždění a ztrátovost paketů byl zvolen výpis nastaveného zpoždění nebo ztrátovosti a také výpis volaného čísla. Mezi příkazy volání a testování pomocí PESQ byl vložen příkaz `sleep 20`, který zajišťuje čekání po dobu 20 sekund a slouží pro ujištění, že volání bude před pesq hodnocením řádně ukončeno. Dále byl také vytvořen jednoduchý textový soubor pro oddělení testovaných vzorků v `pesq_results.txt` souboru. Do textového souboru bylo vloženo oznámení o navýšení zpoždění, ztrátovosti nebo o použití dalšího řečového vzorku. Vkládání bylo vyřešeno pomocí výpisu souboru a následného vložení do `pesq_results.txt` po skončení vnitřního `for` cyklu

```
less soubor.txt >> pesq_results.txt
```

Samotný bash skript je tvořen prvním řádkem

```
#!/bin/bash
```

který zajišťuje jeho spustitelnost pomocí souboru `/bin/bash`, `#` slouží pro komentář oproti tomu `!` uvozuje místo kde skript najde potřebné informace

Skripty jsou tvořeny v textových souborech a jsou zakončeny příponou `.sh`

Příkazy mohou být ve skriptu volně řazeny pod sebe nebo vloženy do příkazu `echo ```.

Pro spuštění musí být skriptu přidělena potřebná práva pomocí:

```
chmod 777 skript.sh
```

samotný skript je poté spuštěn pomocí příkazu

```
./skript.sh
```

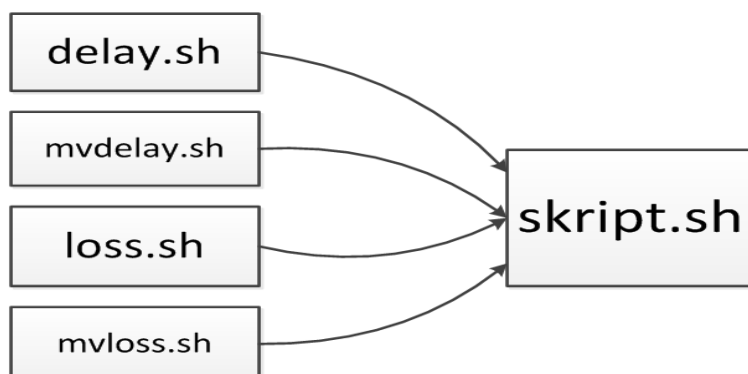
Postup konfigurace viz [14]. Jednotlivé skripty jsou dostupné v příloze a umístěny na CD v textových souborech.

Celé pracoviště bylo ve výsledku koncipováno tak, aby v jednom skriptu byla zajištěna změna zpoždění i ztrátovosti jak je viditelné na Obrázku 4.7. Skripty zůstaly totožné, jen byly doplněny o příkazy přejmenování souboru pesq\_results.txt. Přejmenování je nutné z důvodu ukončení zapisování výsledků do již existujícího souboru a pro vytvoření souboru nového. Toto je prováděno pro větší přehlednost, aby výsledky zpoždění a ztrátovosti byly v jiném textovém souboru. Přejmenování je vytvořeno pomocí:

```
chmod 777 pesq_results.txt
```

```
mv /var/spool/asterisk/monitor/pesq_results.txt
```

```
/var/spool/asterisk/monitor/MOSdelay.txt
```



Obrázek 4.7: Koncepce pracoviště

Výstupem z celého pracoviště jsou textové soubory obsahující získané MOS hodnoty. Soubory jsou předem nakonfigurovány jak je viditelné na Obrázku 4.8. Obsahují název referenčního vzorku, degradovaného vzorku, výslednou MOSPESQ hodnotu a MOS-LQO. Je zde také viditelná vzorkovací frekvence vzorku a zvolené pásmo.

REFERENCE	DEGRADED	PESQMOS	MOSLQO	SAMPLE_FREQ	MODE
M100-out.wav	M100-in.wav	4.466	4.528	8000	nb
M100-out.wav	M100-in.wav	4.466	4.528	8000	nb
M100-out.wav	M100-in.wav	4.449	4.517	8000	nb
M100-out.wav	M100-in.wav	4.449	4.517	8000	nb
M100-out.wav	M100-in.wav	4.368	4.464	8000	nb
M100-out.wav	M100-in.wav	4.368	4.464	8000	nb

Obrázek 4.8: Ukázka výstupních textových souborů

---

## 5 Řečové vzorky

Vzorky byly získávány s ohledem na následné využití pro měření kolegy Radima Feču, jehož úkolem bylo porovnat vliv řeči na výslednou hodnotu MOS. Získali jsme vzorky obsahující ženskou i mužskou řeč a to v českém a anglickém jazyce. Vzorky obsahující českou řeč byly získány z ČVUT a jedná se převážně o nahrávky z rádiového vysílání. Vzorky anglické byly staženy z internetové databáze [7] a [8]. Celkem jsme opatřili 100 vzorků. Vzorky pro každý jazyk a řeč byly určeny v počtu 25. Některé vzorky byly při pokusu o degradaci nezpracovány, protože jejich vzorkovací frekvence nebyla 8000Hz, ale 16000Hz. Proto jsme museli vyřešit změnu vzorkovací frekvence u těchto vzorků. Tento problém byl vyřešen pomocí Linuxového nástroje SOX, jehož bylo nutno nejdříve nainstalovat pomocí

```
apt-get install sox
```

po úspěšné instalaci je třeba povolit u vzorku potřebná práva. Vzorku lze změnit vzorkovací frekvenci pomocí jednoho příkazu

```
chmod 777 M100.wav
```

```
sox M100.wav -b 16 M100.wav channels 1 rate 8k
```

příkaz se skládá z názvu měněného vzorku, pomocí parametru `-b` určíme počet vzorkovacích bitů, název vzorku po změně a parametrem `channels` definujeme počet použitých kanálů `*` = 1,2,... mono, stereo, .... Posledním parametrem `rate` určíme výslednou vzorkovací frekvenci vzorku.

Po změně všech potřebných vzorků a jejich vhodném pojmenování byla celá databáze připravena k použití. Jednotlivé vzorky a jejich výsledné MOS hodnoty jsou dostupné v příloze na CD. Zde jsou také vytvořeny modifikace `extensions.conf` pro jednotlivé sady vzorků.

---

## 6 Závěr

Stanoveným cílem této bakalářské práce byl návrh experimentálního pracoviště pro hodnocení hovoru pomocí intrusivních metod. Tomu ovšem muselo předcházet nastudování dané problematiky hodnocení kvality hovoru, jež bylo zúročeno ve druhé kapitole. Pro realizaci samotného pracoviště bylo třeba porozumět software Asterisk. Pochopit základní nastavení a syntaxi jednotlivých operací. O software Asterisk a jeho provozu pojednává kapitola třetí. Po předešlém prostudování bylo zrealizováno pracoviště, které je založeno na komunikaci mezi dvěma Asterisky, kdy na odchozí rozhraní byly aplikovány různé rozsahy zpoždění a ztrátovosti paketů. Hovor na pracovištích byl nahráván a poté hodnocen pomocí PESQ algoritmu. Smyslem pracoviště byla demonstrace vlivu ztrátovosti a zpoždění na kvalitu hovoru. Tyto vlivy pozoroval a pracoviště nadále užíval kolega Radim Feču pro měření v rámci jeho bakalářské práce. Pracoviště bude nadále využíváno pro experimenty s kvalitou hovoru a pro výuku. Mým osobním přínosem této práci byl návrh koncepce pracoviště a jeho následná realizace. Pracoviště bylo automatizováno pomocí bash skriptů, ve kterých byly definovány jednotlivé operace. Výstupem z celého pracoviště byly textové soubory s výslednými MOS hodnotami.



---

## Použitá literatura

- [1] BRADA, Miloslav a Jan ZELENKA. POSUZOVÁNÍ KVALITY HLASU: Teorie a praxe IP telefonie -3. dvoudenní odborný seminář. [online]. Kongresové centrum Hotelu Olšanka 5. a 6. listopad [cit. 2013-04-30]. Dostupné z: [http://www.ip-telefon.cz/archiv/dok\\_osta/ipt-2008\\_Posuzovani\\_kvality\\_hlasu.pdf](http://www.ip-telefon.cz/archiv/dok_osta/ipt-2008_Posuzovani_kvality_hlasu.pdf)
- [2] VOZŇÁK Miroslav. *Voice over IP*. VŠB-TU Ostrava, 1. vyd., 2008, ISBN 978-80-248-1828-3.
- [3] VOZŇÁK, Miroslav. *Spojovací systémy*. 2. vyd. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2012.
- [4] ITU-T Recommendation P.800. *Methods for objective and subjective assessment of quality*. Geneva, 08/1996. Dostupné z: <http://www.itu.int/rec/T-REC-P.800-199608-I/en>
- [5] ITU-T Recommendation P.800.1. *Mean Opinion Score (MOS) terminology*. Geneva, 07/2006. Dostupné z: <http://www.itu.int/rec/T-REC-P.800.1-200607-I>
- [6] ITU-T Recommendation P.862. *Perceptual evaluation of speech quality (PESQ)*, Geneva, 12/2011. Dostupné z: <http://www.itu.int/rec/T-REC-P.862/>
- [7] Zdroj MA řečových vzorků [online]. [cit. 2013-04-30]. Dostupné z: [http://www.speech.cs.cmu.edu/cmu\\_arctic/packed/](http://www.speech.cs.cmu.edu/cmu_arctic/packed/)
- [8] Zdroj ZA řečových vzorků [online]. [cit. 2013-04-30]. Dostupné z: [www.wavsource.com](http://www.wavsource.com)
- [9] KELLER, Ariane. *Manual to Packet Filtering and netem* [online]. ETH Zurich, 2006 [cit. 2013-04-30]. Dostupné z: <http://tcn.hypert.net/tcmanual.pdf>
- [10] VOZŇÁK, Miroslav. TELEKOMUNIKAČNÍ ÚSTŘEDNY ASTERISK: Teorie a praxe IP telefonie -3. dvoudenní odborný seminář. [online]. Kongresové centrum Hotelu Olšanka 5. a 6. listopad [cit. 2013-04-30]. Dostupné z: [http://www.iptelefon.cz/archiv/dok\\_osta/ipt-2008\\_Telefonni\\_ustredny\\_Asterisk.pdf](http://www.iptelefon.cz/archiv/dok_osta/ipt-2008_Telefonni_ustredny_Asterisk.pdf)
- [11] VALOUŠEK, Ondřej. Asterisk:VoIP ústředna - 2 (konfigurace). [online]. 2006 [cit. 2013-04-30]. Dostupné z: <http://www.abclinuxu.cz/clanky/site/asterisk-voip-ustredna-2-konfigurace>
- [12] DIGIUM, The Asterisk Company. Asterisk Quick Start Guide. [online]. 2012 [cit. 2013-04-30]. Dostupné z: [http://www.asterisk.org/sites/asterisk/files/mce\\_files/documents/asterisk\\_quick\\_start\\_guide.pdf](http://www.asterisk.org/sites/asterisk/files/mce_files/documents/asterisk_quick_start_guide.pdf)
- [13] MADSEN, Leif, Jim VAN MEGGELEN a Russell BRYANT. Asterisk:The Definitive Guide. [online]. 2011 [cit. 2013-04-30]. Dostupné z: [http://asteriskdocs.org/en/3rd\\_Edition/asterisk-book-html-chunk/index.html](http://asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/index.html)

- 
- [14] MILAR, Bohdan. Seriál o Bashi. [online]. 2005 [cit. 2013-04-30]. Dostupné z: <http://www.linuxexpres.cz/praxe/serial-o-bashi>
- [15] HEAD, Acoustic. POLQA Application Guide. [online]. 2012 [cit. 2013-04-30]. Dostupné z: [http://www.head-acoustics.de/downloads/eng/application\\_notes/telecom/Appl\\_Guide\\_POLQA\\_e0.pdf](http://www.head-acoustics.de/downloads/eng/application_notes/telecom/Appl_Guide_POLQA_e0.pdf)
- [16] *Oficiální POLQA stránky* [online]. [cit. 2013-04-30]. Dostupné z: <http://www.polqa.de/>

---

## Seznam příloh

Příloha.A:	Bash skript pro variabilní zpoždění paketů .....	xxxiii
Příloha.B:	Bash skript pro variabilní ztrátovost paketů .....	xxxiii
Příloha.C:	Bash skript varianty druhé pro variabilní zpoždění paketů .....	xxxiii
Příloha.D:	Bash skript varianty druhé pro variabilní ztrátovost paketů .....	xxxiii

Součástí BP/DP je CD/DVD.

Adresářová struktura přiloženého CD/DVD:

- Konfigurační soubory
- Promluvy
  - angM
  - angZ
  - czM
  - czZ
- Výsledky